

Learning where to Attend with Deep Architectures for Image Tracking

Misha Denil¹, Loris Bazzani², Hugo Larochelle³ and Nando de Freitas¹

¹University of British Columbia.

²University of Verona.

³Université de Sherbrooke.

Keywords: Restricted Boltzmann machines, Bayesian optimization, bandits, attention, deep learning, particle filtering, saliency

Abstract

We discuss an attentional model for simultaneous object tracking and recognition that is driven by gaze data. Motivated by theories of perception, the model consists of two interacting pathways, identity and control, intended to mirror the what and where pathways in neuroscience models. The identity pathway models object appearance and performs classification using deep (factored)-Restricted Boltzmann Machines. At each point in time the observations consist of foveated images, with decaying resolution toward the periphery of the gaze. The control pathway models the location, orientation, scale and speed of the attended object. The posterior distribution of these states is estimated with particle filtering. Deeper in the control pathway, we encounter an attentional mechanism that learns to select gazes so as to minimize tracking uncertainty. Unlike in our previous work, we introduce gaze selection strategies which operate in the presence

of partial information and on a continuous action space. We show that a straightforward extension of the existing approach to the partial information setting results in poor performance, and we propose an alternative method based on modeling the reward surface as a Gaussian Process. This approach gives good performance in the presence of partial information and allows us to expand the action space from a small, discrete set of fixation points to a continuous domain.

1 Introduction

Humans track and recognize objects effortlessly and efficiently, exploiting attentional mechanisms (Rensink, 2000; Colombo, 2001) to cope with the vast stream of data. We use the human visual system as inspiration to build a system for simultaneous object tracking and recognition from gaze data. An attentional strategy is learned online to choose fixation points which lead to low uncertainty in the location of the target object. Our tracking system is composed of two interacting pathways. Separation of responsibility is a common feature in models from the computational neuroscience literature, as it is believed to reflect a separation of information processing into ventral and dorsal pathways in the human brain (Ungerleider & Mishkin, 1982; Goodale & Milner, 1992; Olshausen et al., 1993a; O'Reilly, 2010).

The *identity* pathway (ventral) is responsible for comparing observations of the scene to an object template using an appearance model, and on a higher level, for classifying the target object. The identity pathway consists of a three layer deep network. The top layer is a multi-fixation Restricted Boltzmann Machine (RBM) (Larochelle & Hinton, 2010), as shown in Figure 1, which accumulates information from the first hidden layer at consecutive time steps. For the lower layers, we use a (factored)-RBM (Hinton & Salakhutdinov, 2006; Ranzato & Hinton, 2010; Welling et al., 2005; Swersky et al., 2011), but autoencoders (Vincent et al., 2008), sparse coding (Olshausen & Field, 1996; Kavukcuoglu et al., 2009), two-layer ICA (Köster & Hyvärinen, 2007) and convolutional architectures (Lee et al., 2009) could also be adopted.

The *control* pathway (dorsal) is responsible for aligning the object template with the full scene, so the remaining modules can operate independently of the object's position and scale. This pathway is separated into a localization module and a fixation module

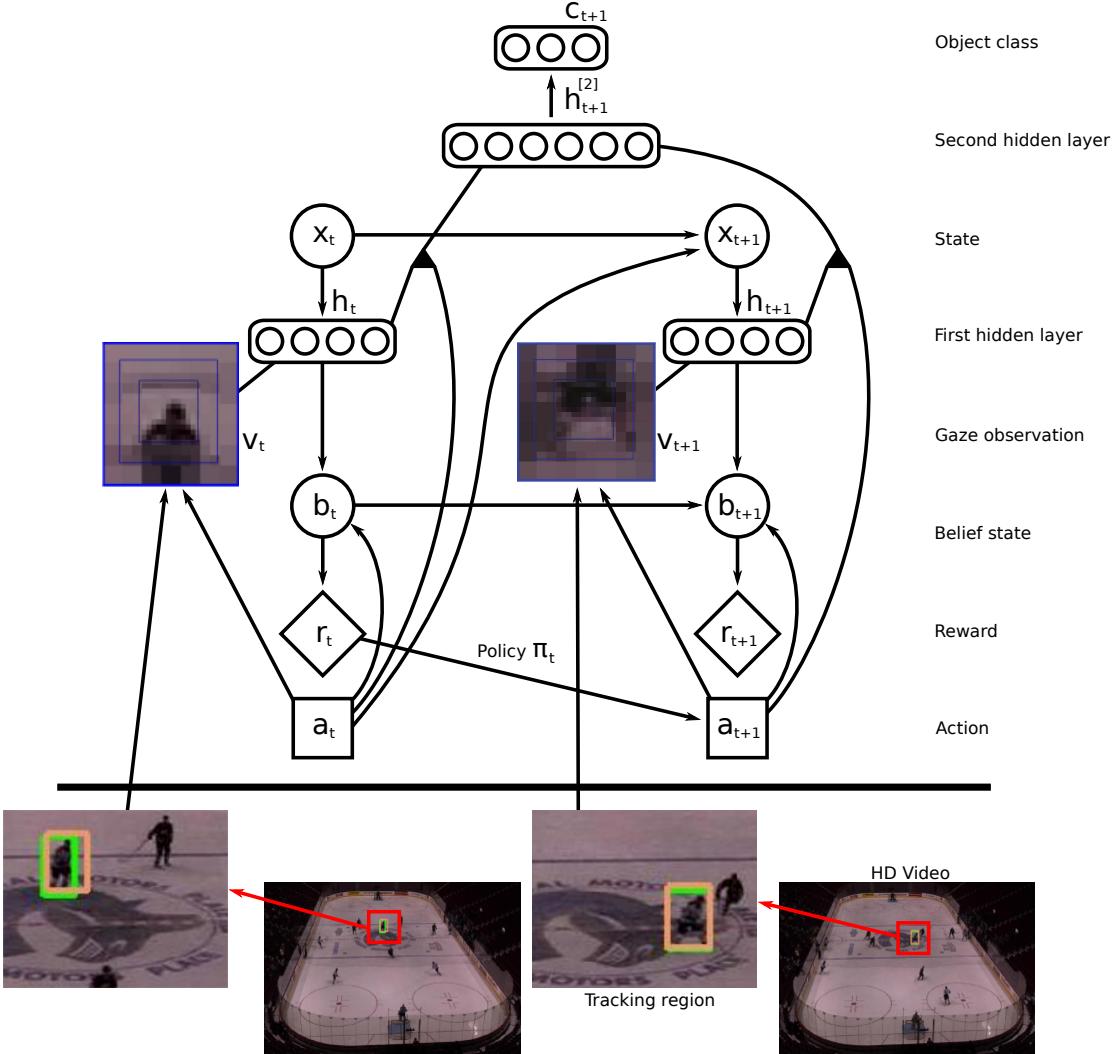


Figure 1: From a sequence of gazes ($\mathbf{v}_t, \mathbf{v}_{t+1}, \dots$), the model infers the hidden features \mathbf{h} for each gaze (that is, the activation intensity of each hidden unit), the hidden features for the fusion of the sequence of gazes $\mathbf{h}^{[2]}$ and the object class \mathbf{c} . The location, size, speed and orientation of the tracking region are encoded in the state \mathbf{x}_t . The actions \mathbf{a}_t follow a learned policy π_t that depends on the past rewards $\{r_1, \dots, r_{t-1}\}$. The immediate reward r_t is a function of the belief state $\mathbf{b}_t = p(\mathbf{x}_t | \mathbf{a}_{1:t}, \mathbf{h}_{1:t})$, also known as the filtering distribution.

which work cooperatively to accomplish this goal. The localization module is implemented as a particle filter (Doucet et al., 2001) which estimates the location, velocity and scale of the target object. We make no attempt to implement such states with neural architectures, but it seems clear that they could be encoded with grid cells (McNaughton et al., 2006) and retinotopic maps as in V1 and the superior colliculus (Rosa, 2002; Girard & Berthoz, 2005). The fixation module learns an attentional strategy to select

fixation points relative to the object template. These fixation points are the centers of partial template observations, and are compared with observations of the corresponding locations in the scene using the appearance model (see Figure 2). Reward is assigned to each fixation based on the uncertainty in the estimate of the target location at each time step. Note that different utilities can be used to reach different goals. Nelson et al. (2010), for example, presented various measures of uncertainty used in decision making for Bayesian optimal-experimental design: probability gain, shannon entropy, Kullback-Leibler distance and impact. In this work, the fixation module uses the reward signal to adapt its fixation policy to achieve good localization; however, by modifying the reward mechanism the fixation policy can be tuned to optimize other objectives, such as maximizing the confidence in classification.

Our previous work (Bazzani et al., 2011) used Hedge (Auer et al., 1998a; Freund & Schapire, 1997) to learn this policy. In this extended paper we show that a straightforward adaptation of our previous approach to the partial information setting results in poor performance, and we propose an alternative method based on modeling the reward surface as a Gaussian Process. This approach gives good performance in the presence of partial information and allows us to expand the action space from a small, discrete set of fixation points to a continuous domain. Videos of our system in action can be found at <http://www.youtube.com/user/anonymousTrack>.

The proposed system can be motivated from many different perspectives. First, starting with Isard & Blake (1996), many particle filters have been proposed for image tracking, but these typically use simple observation models such as B-splines (Isard & Blake, 1996) and colour templates (Okuma et al., 2004). RBMs are more expressive models of shape, and hence we conjecture that they will play a useful role where simple appearance models fail. Second, from a deep learning computational perspective, this work allows us to tackle large images and video, which is typically not possible due to the number of parameters required to represent large images in deep models. The use of fixations synchronized with information about the state (e.g. location and scale) of such fixations eliminates the need to model the entire frame. Third, the system is invariant to image transformations encoded in the state, such as location, scale and orientation. Fourth, from a dynamic sensor network perspective, this paper presents a very simple, but efficient and novel, way of deciding how to gather measurements dynamically.

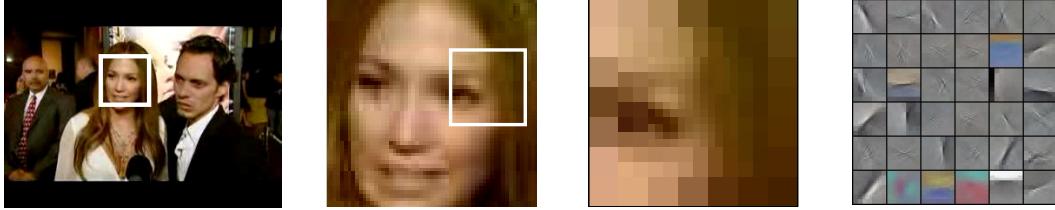


Figure 2: **Left:** A typical HD video frame with the estimated target region highlighted. The target region is estimated at each time step to cope with the large image size. **Centre left:** The template is the target region from the first frame in the video. **Centre right:** To further cope with size only the features of a small region of the template (a fixation) are compared to the features of a small region of the target region. The fixation is foveated with high resolution near the centre and low resolution on the periphery to further reduce the dimensionality. **Right:** The most active features of the first layer (factored)-RBM for the displayed template fixation. The control pathway compares these features to the features active at the corresponding scene location in order to update the belief state.

Lastly, in the context of psychology, the proposed model realizes to some extent the functional architecture for dynamic scene representation of Rensink (2000). The rate at which different attentional mechanisms develop in newborns (including alertness, saccades and smooth pursuit, attention to object features and high-level task driven attention) guided the design of the proposed model and was a great source of inspiration (Colombo, 2001).

Our attentional model can be seen as building a saliency map (Koch & Ullman, 1985) over the target template. Previous work on saliency modeling has focused on identifying salient points in an image using a bottom up process which looks for outliers under some local feature model (which may include top down information in the form of a task dependent prior, global scene features, or various other heuristics). These features can be computed from static images (Torralba et al., 2006; Barrington et al., 2008), or from local regions of spacetime (Gaborski et al., 2004) for video. Additionally, a wide variety of different feature types have been applied to this problem, including engineered features (Gao et al., 2007) as well as features that are learned from data (Zhang et al., 2009). Core to these methods is the idea that saliency is determined by some type of novelty measure. From the computer vision perspective, Lampert et al. (2008) proposed a top-down, branch-and-bound method (faster than standard sliding

window) to select the salient parts of the image for purposes of object detection and localization. Our approach is different, in that rather than identifying locally or globally novel features, our process identifies features which are useful for the task at hand. In our system the saliency signal for a location comes from a top down process which evaluates how well the features at that location enable the system to localize the target object. The work of Gao et al. (2007) considers a similar approach to saliency by defining saliency to be the mutual information between the features at a location and the class label of an object being sought; however, in order to make their model tractable the authors are forced to use specifically engineered features and the approach is tightly coupled to their chosen task. Our system is able to handle arbitrary feature types, and although we consider only on localization in this paper, our model is sufficiently general to be applied to identifying salient features for other goals.

Recently, a dynamic RBM state-space model was proposed in Taylor et al. (2010). Both the implementation and intention behind that proposal are different from the approach discussed here. To the best of our knowledge, our approach is the first successful attempt to combine dynamic state estimation from gazes with online policy learning for gaze adaptation, using deep network network models of appearance. Many other dual-pathway architectures have been proposed in computational neuroscience, including Olshausen et al. (1993b) and Postma et al. (1997), but we believe ours has the advantage that it is very simple, *modular* (with each module easily replaceable), suitable for large datasets and easy to extend.

Another interpretation of this work is as a new model for jointly learning to control eye movements (in smooth pursuit) and to estimate some unknown state of the world. Specifically, in this paper we focus on estimating the position of a moving object. Najaemnik & Geisler (2005) is another example of a model performing estimation and control, but in a visual search task in which the estimation of belief is non-linear and the control policy is greedy. Butko & Movellan (2008) later improved on this work by casting the problem as a partially observed Markov decision process (POMDP) and applying a policy gradient algorithm to perform long term planning, based on an infomax reward. Erez et al. (2011) proposed a slightly different formulation based on continuous state representations (as opposed to a discretized state space), and applied it to the problem of learning hand-eye coordination. Martinez-Cantin et al. (2007) applied Bayesian

optimization to guide policy search so as to minimize the uncertainty in the location of visual features in the setting of a robot exploring and mapping its environment. In their continuous POMDP formulation, the immediate reward is a function of the belief state. Note that in classical POMDP formulations, the immediate reward only depends on the states and actions. Of course, if the POMDP is discrete, it can be mapped to an MDP in belief state space. In doing so, the new reward becomes the expectation of the immediate reward with respect to the belief state (Kaelbling et al., 1998). Vogel & de Freitas (2008) also considered the use of policy search to reduce uncertainty in a gaze planning task with discrete actions. The literature on POMDPs for visual attention is very large. We have simply outlined a few of those works and point the readers to the references in those works. We should state that POMDPs are however very expensive to solve. In tasks where there is abundant data, it seems erroneous to delay the acquisition of such data in favour of completing long term planning calculations. We believe that a myopic strategy coupled with constant data acquisition and the use of this data to construct statistics that allow us to manage a trade off between exploration and exploitation is more natural in data rich domains.

Kanan & Cottrell (2010) proposed another model and applied it to the task of estimating the class of some input image from multiple fixations. Estimation in their model is based on a nonparametric classifier, while control is random and based on a saliency map, derived from a model of natural images. Barrington et al. (2008) proposed an extension of the natural input memory to a Bayesian framework to decide where to saccade to next. Their approach is based on a bottom-up mechanism to build a saliency map that inhibits previously selected fixation points in order to encourage exploration. The acquired image fragments are fused for multi-class recognition using a naïve Bayes classifier. A distinguishing feature of our work is that we applied our model to video, as opposed to still images.

2 Identity Pathway

The identity pathway in our model mirrors the ventral pathway in neuroscience models. It is responsible for modelling the appearance of the target object and also, at a higher level, for classification. More specifically, we opt for a three layer architecture, followed

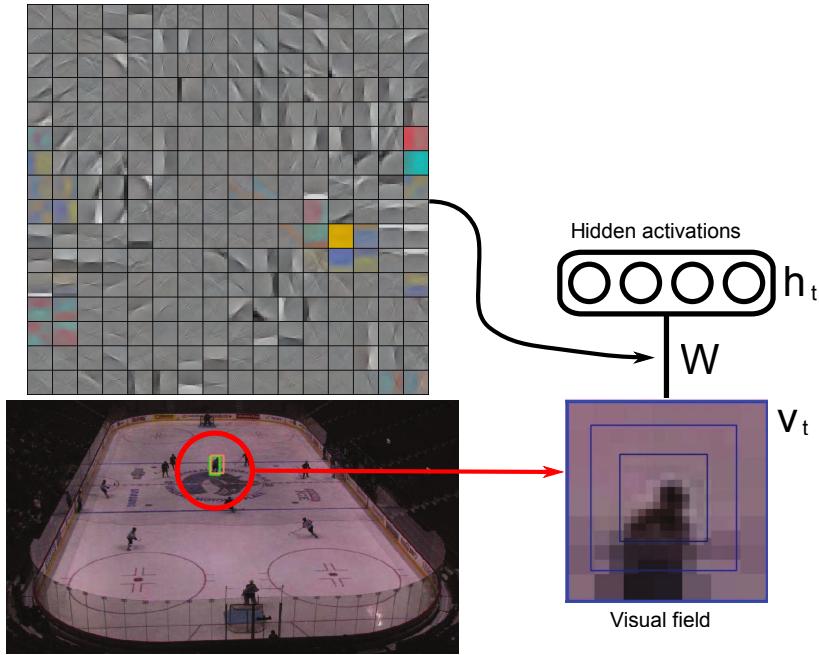


Figure 3: An RBM senses a small foveated image derived from the video. The level of activation of each filter is recorded in the \mathbf{h}_t units. The RBM weights (filters) \mathbf{W} are visualized in the upper left. We currently pre-train these weights.

by a classification module (see Figure 4). The first layer of this pathway is a fixation (Figure 2, centre right) and the second layer aims at modelling the statistics of these fixations. The third layer is trained to combine information about the relative position of many fixations with the first layer activations generated by those fixations, into a coherent representation. Finally, a classifier predicts the category of the tracked object based on the representation computed at the third layer. Each step in this pathway is pre-trained greedily.

2.1 Appearance Model

The second layer varies depending on the type of visual stimuli being modeled. For binary inputs, we use a Restricted Boltzmann Machine (RBM) (Smolensky, 1986; Freund & Haussler, 1991). Noting \mathbf{v}_t as the observed fixation and \mathbf{h}_t the RBM’s binary hidden layer at time t , the energy assigned by the RBM is defined as

$$E(\mathbf{v}_t, \mathbf{h}_t) = -\mathbf{d}^\top \mathbf{h}_t - \mathbf{b}^\top \mathbf{v}_t - \mathbf{h}_t^\top \mathbf{W} \mathbf{v}_t$$

and probabilities are assigned through the Boltzmann distribution

$$p(\mathbf{v}_t, \mathbf{h}_t) = \exp(-E(\mathbf{h}_t, \mathbf{v}_t)) / Z.$$

Given a collection of randomly sampled fixations, the first layer RBM weights \mathbf{W} , and biases \mathbf{d}, \mathbf{b} can be trained using contrastive divergence (Hinton, 2002). We refer the reader to Hinton (2010) for a description of good practices in training RBMs. The end result is a hidden representation of the appearance of individual fixations \mathbf{v}_t ¹, as

$$\mathbf{h}(\mathbf{v}_t) = [p(h_i = 1 | \mathbf{v}_t)]_{i=1}^H = [\text{sigm}(d_i + \mathbf{W}_{i,:}\mathbf{v}_t)]_{i=1}^H$$

where the notation $\mathbf{W}_{i,:}$ refers to the i^{th} row of the matrix \mathbf{W} and H is the number of hidden units of the first layer.

For stimuli better represented with real-valued inputs, such as colour images, we used the factored RBM of Ranzato & Hinton (2010) which is based on a different energy function that can be broken down in two parts:

$$\begin{aligned} E^c(\mathbf{v}_t, \mathbf{h}_t^c) &= -(\mathbf{d}^c)^\top \mathbf{h}_t^c - \sum_{f=1}^F (\mathbf{P}_{f,:}\mathbf{h}_t^c)(\mathbf{C}_{f,:}\mathbf{v}_t)^2 \\ E^m(\mathbf{v}_t, \mathbf{h}_t^m) &= -(\mathbf{d}^m)^\top \mathbf{h}_t^m - (\mathbf{h}_t^m)^\top \mathbf{W}\mathbf{v}_t \end{aligned}$$

where F is the number of linear factors used to model the 3-way interactions between each hidden unit and pair of input units. A factored RBM models the data with two groups of hidden units corresponding to the two energy functions defined above: \mathbf{h}^m models the mean intensity of the each pixel independently, and \mathbf{h}^c captures the pairwise interactions between pixel values. Hybrid Monte Carlo (HMC) can then be used within a similar contrastive divergence procedure to train the parameters of this RBM.

The hidden layer representation is defined using both sets of hidden units $\mathbf{h}(\mathbf{v}_t) = [\mathbf{h}^c(\mathbf{v}_t); \mathbf{h}^m(\mathbf{v}_t)]$, where

$$\begin{aligned} \mathbf{h}^c(\mathbf{v}_t) &= [p(h_i^c = 1 | \mathbf{v}_t)]_{i=1}^{H^c} = \left[\text{sigm} \left(d_i^c + \sum_{f=1}^F P_{f,i} (\mathbf{C}_{f,:}\mathbf{v}_t)^2 \right) \right]_{i=1}^{H^c} \\ \mathbf{h}^m(\mathbf{v}_t) &= [p(h_i^m = 1 | \mathbf{v}_t)]_{i=1}^{H^m} = [\text{sigm}(d_i^m + \mathbf{W}_{i,:}\mathbf{v}_t)]_{i=1}^{H^m}. \end{aligned}$$

¹More specifically, the first layer representation actually depends on the estimated track and the fixation point determined by gaze control (and implicitly, on the whole visual field), which yields the observed fixation \mathbf{v}_t . For simplicity of presentation, and to make our notation more compatible with the RBM literature, we ignore this dependency in the notation for now.

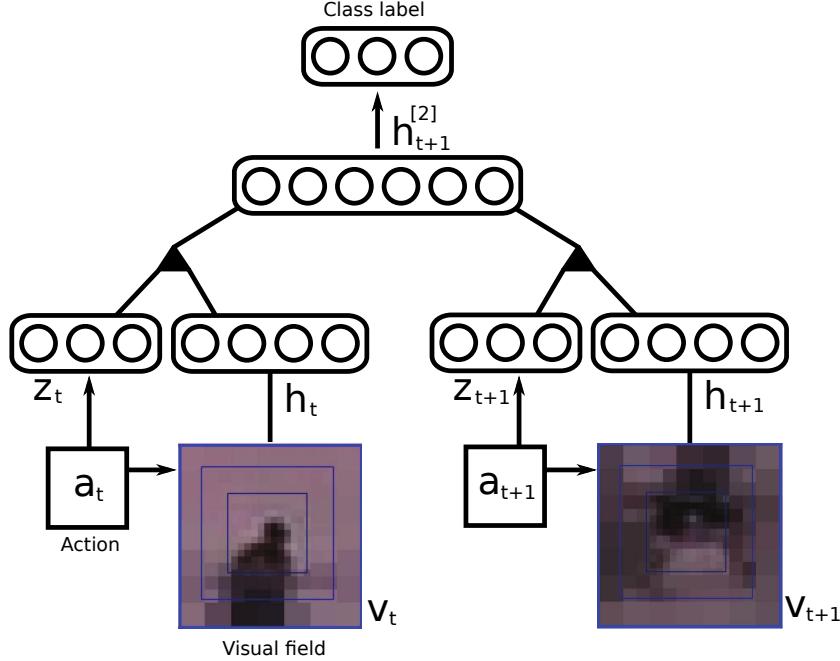


Figure 4: *Gaze accumulation and classification in the identity pathway. A multi-fixation RBM models the conditional distribution (given the gaze positions a_t) of Δ consecutive hidden features h_t , extracted by the first layer RBM (in this illustration, $\Delta = 2$). The multi-fixation RBM encodes the gaze position a_t in a “one hot” representation noted z_t . The activation probabilities of the second layer hidden units $h_t^{[2]}$ are used by a logistic regression classifier to predict the object’s class.*

The first hidden layer can be understood as playing a similar role as the primary visual cortex (V1). In fact, when trained on patches of natural images, the factored RBM learns to extract a representation similar to the Gabor transform V1 neurons seem to be computing (Ranzato & Hinton, 2010). Also, much like the how in many neuroscience models V1 appears in both the ventral and dorsal pathways, the second layer of our identity pathway also appears in the control pathway (see Section 3.1).

2.2 Classification Model

Subsequent steps of the identity pathway are aimed at performing object recognition and classifying a sequence of fixations selected by the fixation policy.

To achieve this, we implemented a multi-fixation RBM very similar to the one proposed in Larochelle & Hinton (2010), where the binary variables z_t (see Figure 4) are

introduced to encode the relative gaze location \mathbf{a}_t within the multi-fixation RBM (a “1 in K ” or “one hot” encoding of the gaze location was used for \mathbf{z}_t). This model sits on top of the appearance model described in the previous section.

The multi-fixation RBM uses the relative gaze location information in order to aggregate the first hidden layer representations \mathbf{h}_t at Δ consecutive time steps into a single, higher level representation $\mathbf{h}_t^{[2]}$. More specifically, the energy function of the multi-fixation RBM is given by

$$E(\mathbf{h}_{t-\Delta+1:t}, \mathbf{z}_{t-\Delta+1:t}, \mathbf{h}_t^{[2]}) = -\mathbf{d}^{[2]\top} \mathbf{h}_t^{[2]} - \sum_{i=1}^{\Delta} \left(\mathbf{b}^{[2]\top} \mathbf{h}_{t-\Delta+i} + \sum_{f=1}^F (\mathbf{P}_{f,:}^{[2]} \mathbf{h}_t^{[2]})(\mathbf{W}_{f,:}^{[2]} \mathbf{h}_{t-\Delta+i})(\mathbf{V}_{f,:}^{[2]} \mathbf{z}_{t-\Delta+i}) \right) .$$

From this energy function, we define a distribution over $\mathbf{h}_{t-\Delta+1:t}$ and $\mathbf{h}_t^{[2]}$ (conditioned on $\mathbf{z}_{t-\Delta+1:t}$) through the Boltzmann distribution

$$p(\mathbf{h}_{t-\Delta+1:t}, \mathbf{h}_t^{[2]} | \mathbf{z}_{t-\Delta+1:t}) = \exp \left(-E(\mathbf{h}_{t-\Delta+1:t}, \mathbf{z}_{t-\Delta+1:t}, \mathbf{h}_t^{[2]}) \right) / Z(\mathbf{z}_{t-\Delta+1:t}) , \quad (1)$$

where the normalization constant $Z(\mathbf{z}_{t-\Delta+1:t})$ ensures that Equation 1 sums to 1. To sample from this distribution, one can use Gibbs sampling by alternating between sampling the top-most hidden layer $\mathbf{h}_t^{[2]}$ given all individual processed gazes $\mathbf{h}_{t-\Delta+1:t}$ and vice versa. To train the multi-fixation RBM, we collect a training set consisting of sequences of Δ pairs $(\mathbf{h}_t, \mathbf{z}_t)$, obtained by randomly selecting Δ fixation points and computing the associated \mathbf{h}_t . These sets are extracted from a collection of images in which the object to detect has been centred. Unsupervised learning using contrastive divergence can then be performed on this training set. See Larochelle & Hinton (2010) for more details.

The main difference between this multi-fixation RBM and the one described in Larochelle & Hinton (2010) is that here $\mathbf{h}_t^{[2]}$ does not explicitly model the class label \mathbf{c}_t . Instead, a multinomial logistic regression classifier is trained separately to predict \mathbf{c}_t from the aggregated representation in $\mathbf{h}_t^{[2]}$. In this way, the multi-fixation RBM can be trained on unlabeled data and thus independently from the recognition task. Specifically, we use the vector of activation probabilities of the hidden units $h_{t,j}^{[2]}$ in $\mathbf{h}_t^{[2]}$,

conditioned on $\mathbf{h}_{t-\Delta+1:t}$ and $\mathbf{z}_{t-\Delta+1:t}$, as the aggregated representation,

$$\begin{aligned} p(h_{t,j}^{[2]} = 1 | \mathbf{h}_{t-\Delta+1:t}, \mathbf{z}_{t-\Delta+1:t}) \\ = \text{sigm} \left(d_j + \sum_{i=1}^{\Delta} \sum_{f=1}^F \mathbf{P}_{f,j}^{[2]} (\mathbf{W}_{f,:}^{[2]} \mathbf{h}_{t-\Delta+i}) (\mathbf{V}_{f,:}^{[2]} \mathbf{z}_{t-\Delta+i}) \right) . \end{aligned}$$

To improve the estimate the class variable \mathbf{c}_t over time, we accumulate the classification decisions at each time step. In particular, the class decision at each time step maximizes over c the probability $p(c|\mathbf{c}_{0:t}) = \frac{\sum_{t'=0}^t I(\mathbf{c}_{t'}=c)}{t}$, where $I(\cdot)$ is the indicator function. We experimented with predicting the class label independently at each time step, but found the multi-fixation module to increase classification accuracy.

Note that the process of pursuit (tracking) is essential to classification. As the target is tracked, the algorithm fixates at locations near the target’s estimated location. The size and orientation of these fixations also depends on the corresponding state estimates. The tracking estimates provide the locations where the algorithm gathers the gazes for classification. Δ gaze positions are randomly selected given the tracking estimates (one for each time step). This random selection is very important when the tracking policy has converged to a specific gaze. In that case, the selected gazes are similar, thus the multi-fixation RBM representation will converge to a single-fixation RBM, decreasing the classification accuracy. It should also be pointed out that instead of using random fixations, one could again use the control strategy proposed in this paper to decide where to look with respect to the track estimate so as to reduce classification uncertainty. We leave the implementation of this extra attentional mechanism for future work.

3 Control Pathway

The control pathway in our model mirrors the responsibility of the dorsal pathway in human visual processing. It tracks the state of the target (position, speed, etc) and normalizes the input so that other modules need not account for these variations. At a higher level, it is responsible for learning an attentional strategy which maximizes the amount of information learned with each fixation. The structure of the control pathway is shown in Figure 5.

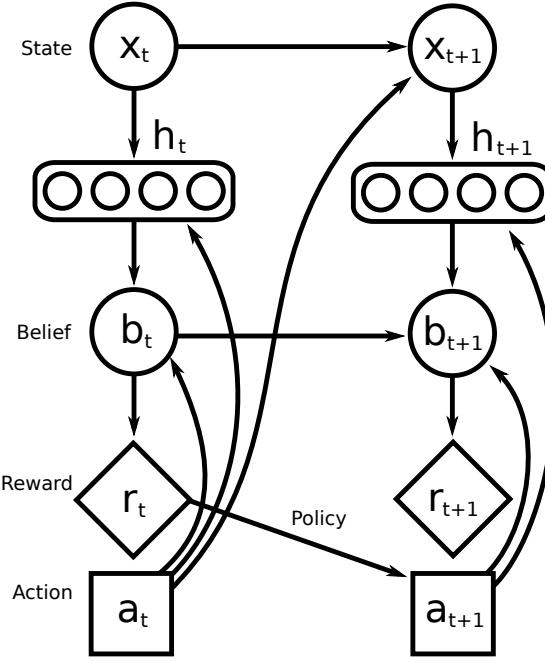


Figure 5: *Influence diagram for the control pathway.* The true state of the tracked object \mathbf{x}_t , generates some set of features \mathbf{h}_t , in the identity pathway. These features depend on the action chosen at time t and are used to update the belief state \mathbf{b}_t . Statistics of the belief state are collected to compute the reward r_t , which is used to update the policy for the next time step.

3.1 State-space model

The standard approach to image tracking is based on the formulation of Markovian, nonlinear, non-Gaussian state-space models, which are solved with approximate Bayesian filtering techniques. In this setting, the unobserved signal (object's position, velocity, scale, orientation or discrete set of operations) is denoted $\{\mathbf{x}_t \in \mathcal{X}; t \in \mathbb{N}\}$. This signal has initial distribution $p(\mathbf{x}_0)$ and transition equation $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1})$. Here $\mathbf{a}_t \in \mathcal{A}$ denotes an action at time t , defined on a compact set \mathcal{A} . For discrete policies \mathcal{A} is finite whereas for continuous policies \mathcal{A} is a region in \mathbb{R}^2 . The observations $\{\mathbf{h}_t \in \mathcal{H}; t \in \mathbb{N}^{>0}\}$, are assumed to be conditionally independent given the process state $\{\mathbf{x}_t; t \in \mathbb{N}\}$. Note that from the state space model perspective the observations are the hidden units of the first hidden layer of the appearance model in the identity pathway.

In summary, the state-space model is described by the following distributions:

$$\begin{aligned} p(\mathbf{x}_0) \\ p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) & \quad \text{for } t \geq 1, \\ p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t) & \quad \text{for } t \geq 1. \end{aligned}$$

For the transition model, we adopt a classical autoregressive process. For the observation model, we follow common practice in image tracking and define it in terms of the distance of the observations from a template τ ,

$$p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t) \propto \exp(-d(\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t), \tau)) ,$$

where $d(\cdot, \cdot)$ denotes a distance metric and τ an object template (for example, a colour histogram or spline). Notice how we have now changed the notation from $\mathbf{h}(\mathbf{v}_t)$ to $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t)$, to emphasize that the hidden unit activations are actually driven by the attentional policy, which in turn generates the fixation \mathbf{v}_t in the first layer RBM.

In this model, the observation $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t)$ is a function of the current state hypothesis and the selected action. The difficulty with this approach is eliciting a good template. Often colour histograms or splines are insufficient. For this reason, we construct a template as follows. First, optical flow is used to detect new object candidates entering the visual scene. Second, we extract a region around the target to use as a visual template, as shown in Figure 2. The same figure also shows a typical foveated observation (higher resolution in the centre and lower in the periphery of the gaze) and the receptive fields for this observation learned beforehand with an RBM. The control algorithm is used to learn which parts of the template are most informative, either by picking from among a predefined set of fixation points, or by using a continuous policy. Finally, we define the likelihood of each observation directly in terms of the distance of the hidden units of the RBM $\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t)$, to the hidden units of the corresponding template region $\mathbf{h}(\mathbf{x}_1, \mathbf{a}_1 = k)$. That is,

$$p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t = k) \propto \exp(-d(\mathbf{h}(\mathbf{x}_t, \mathbf{a}_t = k), \mathbf{h}(\mathbf{x}_1, \mathbf{a}_1 = k))) .$$

The above template is static, but conceivably one could adapt it over time.

Our aim is to estimate recursively in time the *posterior distribution* $p(\mathbf{x}_{0:t} | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ and its associated features, including the marginal distribution $\mathbf{b}_t \triangleq p(\mathbf{x}_t | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ —

known as the *filtering distribution* or *belief state*. This distribution satisfies the following recurrence:

$$\mathbf{b}_t \propto p(\mathbf{h}_t | \mathbf{x}_t, \mathbf{a}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{a}_{t-1}) p(d\mathbf{x}_{t-1} | \mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}) .$$

Except for standard distributions (*e.g.* Gaussian or discrete), this recurrence is intractable. We adopt particle filter to approximate the posterior distribution (see Section 5).

3.2 Reward Function

A gaze control strategy specifies a policy $\pi(\cdot)$ for selecting fixation points. The purpose of this strategy is to select fixation points which maximize an instantaneous reward function $r_t(\cdot)$. The reward can be any desired behaviour for the system, such as minimizing posterior uncertainty or achieving a more abstract goal. We focus on gathering observations so as to minimize the uncertainty in the estimate of the filtering distribution, $r_t(\mathbf{a}_t | \mathbf{b}_t) \triangleq u[\tilde{p}(\mathbf{x}_t | \mathbf{h}_{1:t}, \mathbf{a}_{1:t})]$. More specifically, this reward is a function of the variance of the importance weights w_t of the particle filter approximation of the belief state, that is, $r_t(\mathbf{a}_t | \mathbf{b}_t) = \sum_{i=1}^N (w_t^{(i)})^2$ (see Section 5 for the calculation of the importance weights).

Our particular choice of reward function is motivated by the specific characteristics of our model. Namely,

1. the transition model for the particle filter introduces a lot of diffusion (see Section 5), and
2. our observation model has a very peaked likelihood.

In practice this means that while an object is being successfully tracked only a small fraction of the particles receive any significant weight at each time step, leading to high values for our measure. In contrast, when the system has lost track of its target all the particles will be far from any mode in the likelihood and the particle weights will be approximately uniform, leading to a much lower sum of squared particle weights.

It is also useful to consider the cumulative reward

$$R_T = \sum_{t=1}^T r_t(\mathbf{a}_t | \mathbf{b}_t) ,$$

which is the sum of the instantaneous rewards which have been received up to time T . The gaze control strategies we consider are all “no-regret” which means that the average gap between our cumulative reward and the cumulative reward from always picking the single best action goes to zero as $T \rightarrow \infty$.

In our model each action is a different gaze location and the objective is to choose where to look so as to minimize uncertainty in the belief state.

4 Gaze control

We compare several different strategies for learning the gaze selection policy. In an earlier version of this work (Bazzani et al., 2011) we learned the gaze selection policy with a portfolio allocation algorithm called Hedge (Freund & Schapire, 1997; Auer et al., 1998b). Hedge requires knowledge of the rewards for all actions at each time step, which is not realistic when gazes must be preformed sequentially, since the target object will move between fixations. We compare this strategy (Section 4.2), as well as two baseline methods (Section 4.1), to two very different alternatives (Sections 4.3 and 4.4).

EXP3 is an extension of Hedge to partial information games (Auer et al., 2001). Unlike Hedge, EXP3 requires knowledge of the reward only for the action selected at each time step. EXP3 is more appropriate to the setting at hand, and is also more computationally efficient than Hedge; however, this comes at a cost of substantially lower theoretical performance guarantees.

Both Hedge and EXP3 learn gaze selection policies which choose among a discrete set of predetermined fixation points. We can instead learn a continuous policy by estimating the reward surface using a Gaussian Process (Rasmussen & Williams, 2006). By assuming that the reward surface is smooth, we can draw on the tools of Bayesian optimization (Brochu et al., 2009) to search for the optimal gaze location using as few exploratory steps as possible.

The following sections describe each of these approaches in more detail.

4.1 Baseline

We consider two baseline strategies, which we call random and circular. The random strategy samples fixation points uniformly at random from a small discrete set of possibilities. The circular strategy also uses a small discrete set of fixation points and cycles through them in a fixed order.

4.2 Hedge

To use Hedge (Freund & Schapire, 1997; Auer et al., 1998b) for gaze selection we must first discretize the action space by selecting a fixed finite number of possible fixation points. Hedge maintains an importance weight $G(i)$ for each possible fixation point and uses them to form a stochastic policy at each time step. An action is selected according to this policy and the reward for each possible action is observed. These rewards are then used to update the importance weights and the process repeats. Pseudo code for Hedge is shown in Algorithm 1.

Algorithm 1 Hedge

Input: $\gamma > 0$

Input: $G_0(i) \leftarrow 0$ **foreach** $i \in \mathcal{A}$

for $t = 1, 2, \dots$ **do**

for $i \in \mathcal{A}$ **do**

$$p_t(i) \leftarrow \frac{\exp(\gamma G_{t-1}(i))}{\sum_{j \in \mathcal{A}} \exp(\gamma G_{t-1}(j))}$$

$\mathbf{a}_t \sim (p_t(1), \dots, p_t(|\mathcal{A}|))$ // sample an action from the distribution $(p_t(k))$

for $i \in \mathcal{A}$ **do**

$$r_t(i) \leftarrow r_t(i|\mathbf{b}_t)$$

$$G_t(i) \leftarrow G_{t-1}(i) + r_t(i)$$

4.3 EXP3

EXP3 (Auer et al., 2001) is a generalization of Hedge to the partial information setting. In order to maintain estimates for the importance weights, Hedge requires reward information for each possible action at each time step. EXP3 works by wrapping Hedge in an outer loop which simulates a fully observed reward vector at each time step. EXP3

selects actions based on a mixture of the policy found by Hedge and a uniform distribution. EXP3 is able to function in the presence of partial information, but this comes at the cost of substantially worse theoretical guarantees. Pseudo code for EXP3 is shown in Algorithm 2.

Algorithm 2 EXP3

Input: $\gamma \in (0, 1]$

Initialize **Hedge**(γ)

for $t \in 1, 2, \dots$ **do**

 Receive \mathbf{p}_t from **Hedge**

$$\hat{\mathbf{p}}_t \leftarrow (1 - \gamma)\mathbf{p}_t + \frac{\gamma}{|\mathcal{A}|}$$

$$\mathbf{a}_t \sim (\hat{p}_t(1), \dots, \hat{p}_t(|\mathcal{A}|))$$

 Simulate a reward vector for **Hedge** where $\hat{r}_t(j) \leftarrow \begin{cases} r_t(j)/p_t(j) & \text{if } j = \mathbf{a}_t \\ 0 & \text{otherwise} \end{cases}$

4.4 Bayesian Optimization

Both Hedge and EXP3 discretize the space of possible fixation points and learn a distribution over this finite set. In contrast, Bayesian optimization is able to treat the space as fully continuous by placing a smoothness prior on how reward is expected to vary with location. Intuitively, if we know the reward at one location, then we expect other, nearby locations to produce similar rewards. Gaussian Process priors encode this type of belief (Rasmussen & Williams, 2006), and have been used extensively for optimization of cost functions when it is important to minimize the total number of function evaluations (Brochu et al., 2009).

We model the latent reward function $r_t(\mathbf{a}_t | \mathbf{b}_t) \triangleq r(\mathbf{a}_t | \mathbf{b}_t, \boldsymbol{\theta}_t)$ as a zero mean Gaussian Process

$$r(\mathbf{a}_t | \mathbf{b}_t, \boldsymbol{\theta}_t) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{a}_t, \mathbf{a}'_t | \mathbf{b}_t, \boldsymbol{\theta}_t)) ,$$

where \mathbf{b}_t is the belief state (see Section 3.1), and $\boldsymbol{\theta}_t$ are the model hyperparameters. The kernel function $k(\cdot, \cdot)$, gives the covariance between the reward at any two gaze locations. To ease the notation, the explicit dependence of $r(\cdot)$ and $k(\cdot, \cdot)$ on \mathbf{b}_t and $\boldsymbol{\theta}_t$ will be dropped.

We assume that the true reward function $r(\cdot)$ is not directly measurable, and what we observe are measurements of this function corrupted by Gaussian noise. That is, at each time step the instantaneous reward r_t , is given by

$$r_t = r(\mathbf{a}_t) + \sigma_n \delta_n ,$$

where $\delta_n \sim \mathcal{N}(0, 1)$ and σ_n is a hyperparameter indicating the amount of observation noise, which we absorb into θ_t .

Given a set of observations we can compute the posterior predictive distribution for $r(\cdot)$:

$$\begin{aligned} r(\mathbf{a} | \mathbf{r}_{1:t}, \mathbf{a}_{1:t}) &\sim \mathcal{N}(m_t(\mathbf{a}), s_t^2(\mathbf{a})) , \\ m_t(\mathbf{a}) &= \mathbf{k}^T [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{r}_{1:t} , \\ s_t^2(\mathbf{a}) &= k(\mathbf{a}, \mathbf{a}) - \mathbf{k}^T [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k} , \end{aligned} \quad (2)$$

where

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} k(\mathbf{a}_1, \mathbf{a}_1) & \cdots & k(\mathbf{a}_1, \mathbf{a}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{a}_t, \mathbf{a}_1) & \cdots & k(\mathbf{a}_t, \mathbf{a}_t) \end{bmatrix} , \\ \mathbf{k} &= \begin{bmatrix} k(\mathbf{a}_1, \mathbf{a}) & \cdots & k(\mathbf{a}_t, \mathbf{a}) \end{bmatrix}^T , \\ \mathbf{r}_{1:t} &= \begin{bmatrix} r_1 & \cdots & r_t \end{bmatrix}^T . \end{aligned}$$

It remains to specify the form of the kernel function, $k(\cdot, \cdot)$. We experimented with several possibilities, but found that the specific form of the kernel function is not critical to the performance of this method. For the experiments in this paper we used the squared exponential kernel,

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma_m^2 \exp \left(-\frac{1}{2} \sum_{k=1}^D \left(\frac{a_{i,k} - a_{j,k}}{\ell_k} \right)^2 \right) ,$$

where σ_m^2 and the $\{\ell_1, \dots, \ell_D\}$ are hyperparameters.

Equation 2 is a Gaussian Process estimate of the reward surface and can be used to select a fixation point for the next time step. The estimate gives both a predicted reward value and an associated uncertainty for each possible fixation point. This is the strength of Gaussian Processes for this type of optimization problem, since the predictions can

be used to balance exploration (choosing a fixation point where the reward is highly uncertain) and exploitation (choosing a point we are confident will have high reward).

There are many selection methods available in the literature which offer different tradeoffs between these two criteria (Hoffman et al., 2011). In this paper we use GP-UCB (Srinivas et al., 2010) which selects

$$\mathbf{a}_{t+1} = \arg \max_{\mathbf{a}} m_t(\mathbf{a}) + \sqrt{\beta_t} s_t(\mathbf{a}) \quad (3)$$

where β_t is a parameter. The setting $\beta_t = 2 \log(t^3 \pi^2 / 3\delta)$ (with $\delta = 0.001$) is used throughout this paper.

Equation 3 must still be optimized to find \mathbf{a}_{t+1} , which can be performed using standard global optimization tools. We use DIRECT (Jones et al., 1993) due to the existence of a readily available implementation.

The Gaussian Process regression is controlled by several hyperparameters (see Figure 6): σ_m^2 controls the overall magnitude of the covariance, and σ_n^2 controls the amount of observation noise. The remaining parameters $\{\ell_1, \dots, \ell_D\}$ are length scale parameters which control the range of the covariance effects in each dimension.

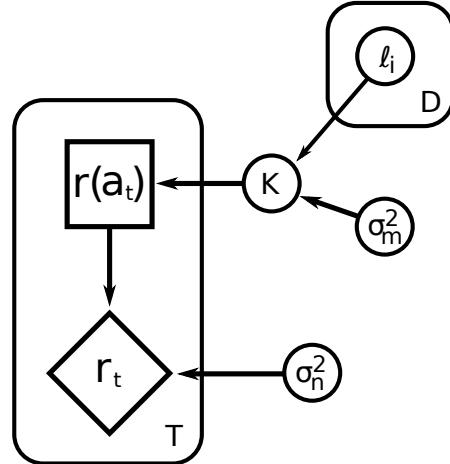


Figure 6: *Graphical model for Bayesian optimization. The ℓ_i are length scales in each dimension, σ_m^2 is the magnitude parameter and σ_n^2 is the noise level. In our model σ_m^2 and σ_n^2 follow a uniform prior and the ℓ_i follow independent Student-t priors.*

Treatment of the hyperparameters requires special consideration in this setting. The pure Bayesian approach is to put a prior on each parameter and integrate them out of the predictive distribution. However, since the integrals involved are not tractable

analytically, this requires computationally expensive numerical approximations. Speed is an issue here since GP-UCB requires that we optimize a function of the posterior process at each time step so, for instance, computing Monte Carlo averages for each evaluation of Equation 2 is prohibitively slow.

An alternative approach is to choose parameter values via maximum likelihood. This can be done quickly, and allows us to make speedy predictions; however, in this case we suffer from problems of data scarcity, especially early in the tracking process when few observations have been made. The length scale parameters are particularly prone to receiving very poor estimates when there is little data available.

We have found that using informative priors for the length scale parameters and making MAP, rather than ML, estimates at each time step provides a solution to the problems described above. MAP estimates can be made quickly using gradient optimization methods (Rasmussen & Williams, 2006), and informative priors provide resistance to the problems encountered with ML. The experiments in Section 7 place uniform priors on the magnitude and noise parameters and place independent Student-t priors on each length scale parameter. The experiments also use an initial data collection phase of 10 time steps before any adjustment of the parameters is made.

5 Belief Updates

Since the belief state cannot be computed analytically, we will adopt particle filtering to approximate it. The full algorithm is shown in Algorithm 3.

We refer readers to Doucet et al. (2001) for a more in depth treatment of these sequential Monte Carlo methods. Assume that at time $t - 1$ we have $N \gg 1$ particles (samples) $\{\mathbf{x}_{0:t-1}^{(i)}\}_{i=1}^N$ distributed according to $p(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1})$. We can approximate this belief state with the following empirical distribution

$$\hat{p}(d\mathbf{x}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}) \triangleq \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_{0:t-1}^{(i)}}(d\mathbf{x}_{0:t-1}) .$$

Particle filters combine sequential importance sampling with a selection scheme designed to obtain N new particles $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^N$ distributed approximately according to $p(d\mathbf{x}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$.

Algorithm 3 Particle filtering algorithm with gaze control. The algorithm shown here is for partial information strategies. For full information strategies the importance sampling step is done independently for each possible action and the gaze control step is able to use reward information from each possible action to create the new strategy $\pi_{t+1}(\cdot)$.

1. Initialization

```

for  $i = 1$  to  $N$  do
     $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$ 

```

Initialize the policy $\pi_1(\cdot)$ // How this is done depends on the control strategy

```
for  $t = 1 \dots$  do
```

2. Importance sampling

```

for  $i = 1$  to  $N$  do                                // Predict the next state

```

$$\begin{aligned}\widetilde{\mathbf{x}}_t^{(i)} &\sim q_t \left(d\mathbf{x}_t^{(i)} \middle| \widetilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t} \right) \\ \widetilde{\mathbf{x}}_{0:t}^{(i)} &\leftarrow \left(\mathbf{x}_{0:t-1}^{(i)}, \widetilde{\mathbf{x}}_t^{(i)} \right)\end{aligned}$$

$k^* \sim \pi_t(\cdot)$ // Select an action according to the policy

```

for  $i = 1$  to  $N$  do                                // Evaluate the importance weights

```

$$\widetilde{w}_t^{(i)} \leftarrow \frac{p \left(\mathbf{h}_t | \widetilde{\mathbf{x}}_t^{(i)}, \mathbf{a}_t = k^* \right) p \left(\widetilde{\mathbf{x}}_t^{(i)} | \widetilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{a}_{t-1} \right)}{q_t \left(\widetilde{\mathbf{x}}_t^{(i)} \middle| \widetilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t} \right)}$$

```

for  $i = 1$  to  $N$  do                                // Normalize the importance weights

```

$$w_t^{(i)} \leftarrow \frac{\widetilde{w}_t^{(i)}}{\sum_{j=1}^N \widetilde{w}_t^{(j)}}$$

3. Gaze control

$r_t = \sum_{i=1}^N (w_t^{(i)})^2$ // Receive reward for the chosen action

Incorporate r_t into the policy to create $\pi_{t+1}(\cdot)$

4. Selection

Resample with replacement N particles $(\mathbf{x}_{0:t}^{(i)}; i = 1, \dots, N)$ from the set $(\widetilde{\mathbf{x}}_{0:t}^{(i)}; i = 1, \dots, N)$ according to the normalized importance weights $w_t^{(i)}$

5.1 Importance sampling step

The joint distributions $p(dx_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1})$ and $p(dx_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ are of different dimension. We first modify and extend the current paths $\mathbf{x}_{0:t-1}^{(i)}$ to obtain new paths $\tilde{\mathbf{x}}_{0:t}^{(i)}$ using a proposal kernel $q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{x}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})$. As our goal is to design a sequential procedure, we set

$$q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{x}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = \delta_{\mathbf{x}_{0:t-1}}(d\tilde{\mathbf{x}}_{0:t-1}) q_t(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) ,$$

that is $\tilde{\mathbf{x}}_{0:t} = (\mathbf{x}_{0:t-1}, \tilde{\mathbf{x}}_t)$. The aim of this kernel is to obtain new paths whose distribution

$$q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = p(d\tilde{\mathbf{x}}_{0:t-1}|\mathbf{h}_{1:t-1}, \mathbf{a}_{1:t-1}) q_t(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t}) ,$$

is as “close” as possible to $p(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$. Since we cannot choose $q_t(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = p(d\tilde{\mathbf{x}}_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ because this is the quantity we are trying to approximate in the first place, it is necessary to weight the new particles so as to obtain consistent estimates.

We perform this “correction” with importance sampling, using the weights

$$\tilde{w}_t = \tilde{w}_{t-1} \frac{p(\mathbf{h}_t|\tilde{\mathbf{x}}_t, \mathbf{a}_t) p(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{a}_{t-1})}{q_t(d\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{0:t-1}, \mathbf{h}_{1:t}, \mathbf{a}_{1:t})} .$$

The choice of the transition prior as proposal distribution is by far the most common one. In this case, the importance weights reduce to the expression for the likelihood. However, it is possible to construct better proposal distributions, which make use of more recent observations, using object detectors (Okuma et al., 2004), saliency maps (Itti et al., 1998), optical flow, and approximate filtering methods such as the unscented particle filter. One could also easily incorporate strategies to manage data association and other tracking related issues. After normalizing the weights, $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}}$, we obtain the following estimate of the filtering distribution:

$$\tilde{p}(dx_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t}) = \sum_{i=1}^N w_t^{(i)} \delta_{\tilde{\mathbf{x}}_{0:t}^{(i)}}(dx_{0:t}) .$$

Finally a selection step is used to obtain an “unweighted” approximate empirical distribution $\hat{p}(dx_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$ of the weighted measure $\tilde{p}(dx_{0:t}|\mathbf{h}_{1:t}, \mathbf{a}_{1:t})$. The basic idea is to discard samples with small weights and multiply those with large weights. The use of a selection step is key to making the SMC procedure effective; see Doucet et al. (2001) for details on how to implement this black box routine.

6 Algorithm

In this section we give a brief overview of how each of the pieces described in the preceding sections fit together in the full system. The content of this section is complementary to the graphical description from Figure 1 in the introduction. Figure 1 illustrates the influence structure between the components of the system, whereas the purpose of this section is to give a more operational description. Below we describe, step by step, how control flows through the model at each time step. The description here is very high level, and refer the reader to previous sections of this paper for the details of each step.

1. Offline, learn the appearance and classification models.
2. For a given video, locate the target object either using an oracle or through optical flow and extract a region of pixels, centred on the target, to use as the object template. Initialize the fixation policy and the particle filter.
3. For a given frame, use the fixation policy to select a fixation point which gives an offset relative to the object template.
4. **Partial information policies:** Each particle in the particle filter gives an alignment of the template with the scene. For each particle, acquire a fixation from the scene corresponding to the fixation point. Since each particle may align the template and the scene differently these fixations will not all be the same. Use the appearance model to compute features for each of these fixations.
Full information policies: For each particle and for each possible fixation point, acquire a fixation from the scene. Use the appearance model to compute features for each of these fixations.
5. Compare the features from each fixation with features generated by the appearance model when observing the fixation point on the template. Use this information to update the weights of the particle filter.
6. Use the new particle filter weights to compute the reward for the attentional strategy. Use this information to compute a new fixation policy for the next time step.

7. Resample the particles in the particle filter.
8. Given the target estimate, feed the features from a randomly selected fixation into the classification model to be incorporated with features from previous time steps. Output a classification prediction.
9. Advance to the next frame and continue from step 3.

7 Experiments

In this section we report the results of running our system on several different video sequences including both synthetic and real world data. We first consider the full information scenario and demonstrate that a learned attentional policy outperforms the baseline strategies both in terms of tracking performance as well as classification accuracy. We then consider learning attentional strategies with partial information. We show that a straightforward generalization of our full information approach to this setting does not perform well, but the Bayesian optimization policy described in Section 4.4 is competitive with our full information approach, even in this more difficult setting.

7.1 Full Information Policies

In this section, three experiments are carried out to evaluate quantitatively and qualitatively the proposed approach. The first experiment provides comparisons between Hedge and the baseline policies. The second experiment, on a similar synthetic dataset, demonstrates how the approach can handle large variations in scale, occlusion and multiple targets. The final experiment is a demonstration of tracking and classification performance on several real videos. For the synthetic digit videos, we trained the first-layer RBMs on the foveated images, while for the real videos we trained factored-RBMs on foveated natural image patches (Ranzato & Hinton, 2010).

The first experiment uses 10 video sequences (one for each digit) built from the MNIST dataset. Each sequence contains a moving digit and static digits in the background (to create distractions). The template had $K = 9$ gaze positions, chosen so that gaze G5 was at the centre as shown in Figure 7, and the objective is to track and recog-

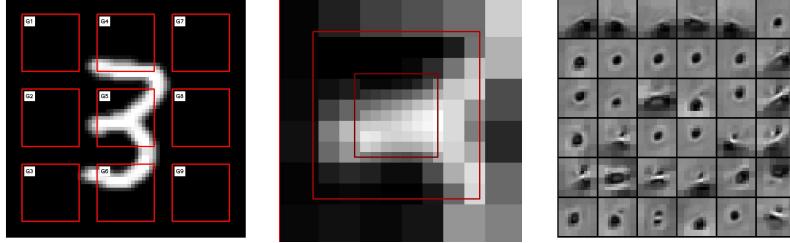


Figure 7: **Left:** An example of a digit template used in the first experiment. The red boxes show the positions of each possible fixation point. **Centre:** The foveated fixation that corresponds to the fixation point G5. **Right:** The most active RBM filters when fixating on G5.

nize the moving digit (see Figure 8). The location of the template was initialized with optical flow.

We compare the leaned policy (Hedge) against two baselines: the random policy and the circular policy (see Section 4.1). The Bhattacharyya distance has been used in the specification of the observation model. A multi-fixation RBM was trained to map the first layer hidden units of three consecutive time steps into a second hidden layer, and we trained a logistic regressor to further map to the 10 digit classes. We used the transition prior as proposal for the particle filter.

Tables 1 and 2 report the comparison results. Tracking accuracy was measured in terms of the mean and standard deviation (in brackets) over time of the distance between the target ground truth and the estimate; measured in pixels. The analysis highlights that the error of the Hedge policy is always below the error of the other policies. In most of the experiments, the tracker fails when an occlusion occurs for the deterministic and the random policies, while the learned policy is successful. This is very clear in the videos at: <http://www.youtube.com/user/anonymousTrack>

The loss of track for the simple policies is mirrored by the high variance results in Table 1 (experiments 0, 1, 4, and so on). The average mean and standard deviations (last column of Table 1) make it clear that the proposed strategy for learning a gaze policy can be of enormous benefit. The improvements in tracking performance are mirrored by improvements in classification performance.

Figure 8 provides further anecdotal evidence for the policy learning algorithm. The top sequence shows the target and the particle filter estimate of its location over time.

	0	1	2	3	4	5	6	7	8	9	Avg
Hedge	1.2 (1.2)	3.0 (2.0)	2.9 (1.0)	2.2 (0.7)	1.0 (1.9)	1.8 (1.9)	3.8 (1.0)	3.8 (1.5)	1.5 (1.7)	3.8 (2.8)	2.5 (1.6)
Circular	18.2 (29.6)	536.9 (395.6)	104.4 (69.7)	2.9 (2.2)	201.3 (113.4)	4.6 (4.0)	5.6 (3.1)	64.4 (45.3)	142.0 (198.8)	144.6 (157.7)	122.5 (101.9)
Random	41.5 (54.0)	410.7 (329.4)	3.2 (2.0)	3.3 (2.4)	42.8 (60.9)	6.5 (9.6)	5.7 (3.2)	80.7 (48.6)	38.9 (50.6)	225.2 (241.6)	85.9 (80.2)

Table 1: *Tracking error (in pixels) on several video sequences using different policies for gaze selection.*

	0	1	2	3	4	5	6	7	8	9	Avg
Hedge	95.62%	100.00%	99.66%	99.33%	99.66%	100.00%	100.00%	98.32%	97.98%	89.56%	98.01%
Circular	99.33%	100.00%	98.99%	94.95%	5.39%	98.32%	0.00%	29.63%	52.19%	0.00%	57.88%
Random	98.32%	100.00%	96.30%	99.66%	29.97%	96.30%	89.56%	22.90%	12.79%	13.80%	65.96%

Table 2: *Classification accuracy on several video sequences using different policies for gaze selection.*

The middle sequence illustrates how the policy changes over time. In particular, it demonstrates that hedge can effectively learn where to look in order to improve tracking performance (we chose this simple example as in this case it is obvious that the centre of the eight (G5) is the most reliable gaze action). The classification results over time are shown in the third row.

The second experiment addresses a similar video sequence, but tracking multiple targets. The image scale of each target changes significantly over time, so the algorithm has to be invariant with respect to these scale transformations. In this case, we used a mixture proposal distribution consisting of motion detectors and the transition prior. We also tested a saliency proposal but found it to be less effective than the motion detectors for this dataset. Figure 9 (top) shows some of the video frames and tracks. The videos allow one to better appreciate the performance of the multi-target tracking algorithm in the presence of occlusions.

Tracking and classification results for the real videos are shown in Figure 9 and the accompanying videos. We analyzed three different scenario: *hockey*, *surveillance*, *face*. The *hockey* scenario consists on a video of hockey players taken from a static camera. For the *surveillance* scenario, we extracted a video from popular public dataset for people detection and tracking, CAVIAR². For the *face* scenario, we use the Youtube celebrity dataset from Kim et al. (2008). This data set consists of several videos of

²<http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

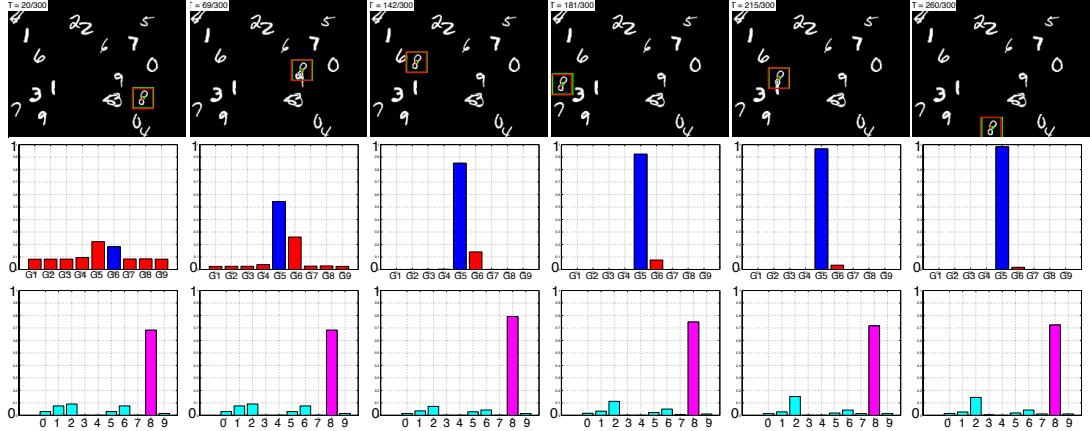


Figure 8: *Tracking and classification accuracy results with the learned policy.* **First row:** Position of the target and estimate over time. **Second row:** Policy distribution over the 9 gazes; Hedge appears to converge to a reasonable policy. **Third row:** Cumulative class distribution for recognition.

celebrities taken from Youtube and is challenging for tracking algorithms as the videos exhibit a wide variety of illuminations, expressions and face orientations. The qualitative results shown in Figure 9 provides additional evidence that the proposed model is able to perform tracking on real data.

7.2 Partial Information Policies

In this section, two experiments are carried out to evaluate the performance between full information and partial information policies.

In the first experiment we compare the performance of each gaze selection method on a data set of several videos of digits from the MNIST data set moving on a black background. The target in each video encounters one or more partial occlusions which the tracking algorithm must handle gracefully. Additionally, each video sequence has been corrupted with 30% noise. We measure the error between the estimated track and the ground truth for each gaze selection method, and demonstrate that Bayesian optimization preforms comparably to Hedge, but that EXP3 is not able to reach a satisfactory level of performance. We also demonstrate qualitatively that the Bayesian optimization approach learns good gaze selection policies on this data set.

Table 3 reports the results from the first experiment. The table shows the mean

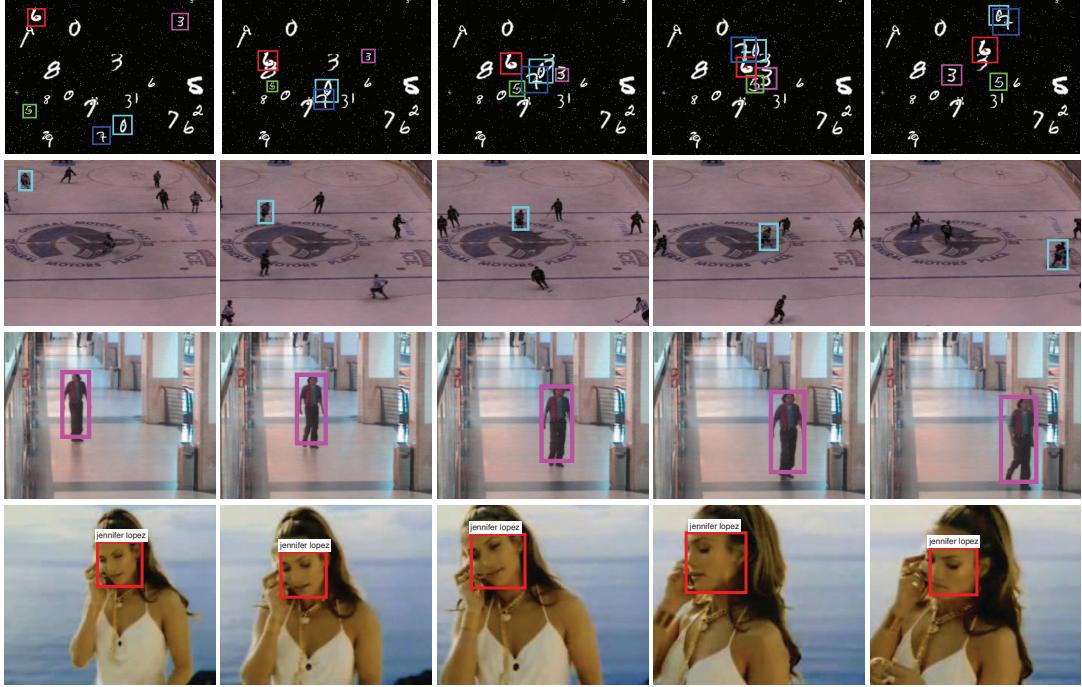


Figure 9: **Top:** Multi-target tracking with occlusions and changes in scale on a synthetic video. **Middle and bottom:** Tracking in real video sequences.

tracking error, measured by averaging distance between the estimated and ground truth track over the entire video sequence. Here we see that the Bayesian optimization approach compares favorably to Hedge in terms of tracking performance, and that EXP3 preforms substantially worse than the other two methods. Although Hedge preforms marginally better than Bayesian optimization, it is important to remember that Bayesian optimization solves a significantly more difficult problem. Hedge relies on discretizing the action space, and must have access to the rewards for all possible actions at each time step. In contrast, Bayesian optimization considers a fully continuous action space, and receives reward information only for the chosen actions.

Figure 10 shows the reward surfaces learned for each digit by Bayesian optimization, as well as a visualization of the overall best fixation points using data aggregated across ten runs. The optimal fixation points found by the algorithm are tightly clustered, and the resulting observations are very distinguishable.

In our second experiment we use the Youtube celebrity dataset. We run our tracking model using Bayesian optimization to learn a gaze selection policy on this data set, and present some results in Figure 11. Although we report only qualitative results from this

	0	1	2	3	4	5	6	7	8	9	Avg
Bayesopt	5.36 (2.32)	7.92 (2.52)	2.62 (3.89)	4.05 (1.67)	1.70 (5.10)	8.31 (3.35)	4.94 (2.28)	12.09 (3.53)	1.52 (2.76)	9.06 (1.66)	5.76 (2.91)
Hedge	2.97 (1.56)	3.20 (2.19)	2.97 (1.99)	2.92 (2.00)	3.14 (1.80)	2.96 (2.08)	2.86 (1.96)	2.98 (1.76)	2.81 (1.64)	3.15 (3.73)	3.00 (2.07)
EXP3	3.18 (5.05)	3.03 (10.08)	65.46 (3212.16)	91.81 (3671.66)	2.62 (2.35)	7.20 (303.29)	67.54 (2346.82)	2.97 (3.99)	3.06 (2.71)	77.01 (3135.17)	32.39 (1269.33)

Table 3: *Tracking error on several video sequences using different methods for gaze selection.* The table shows mean tracking error as well as the error variance (in brackets) over a single test sequence.

experiment, it provides anecdotal evidence that Bayesian optimization is able to form a good gaze selection policy on real world data.

8 Conclusions and Future Work

We have proposed a decision-theoretic probabilistic graphical model for joint classification, tracking and planning. The experiments demonstrate the significant potential of this approach. We examined several different strategies for gaze control in both the full and partial information settings. We saw that a straightforward generalization of the full information policy to partial information gave poor performance and we proposed an alternative method which is able not only to perform well in the presence of partial information and also allows us to expand the set of possible fixation points to a continuous domain.

There are many routes for further exploration. In this work we pre-trained the appearance model. However, existing particle filtering and stochastic optimization algo-

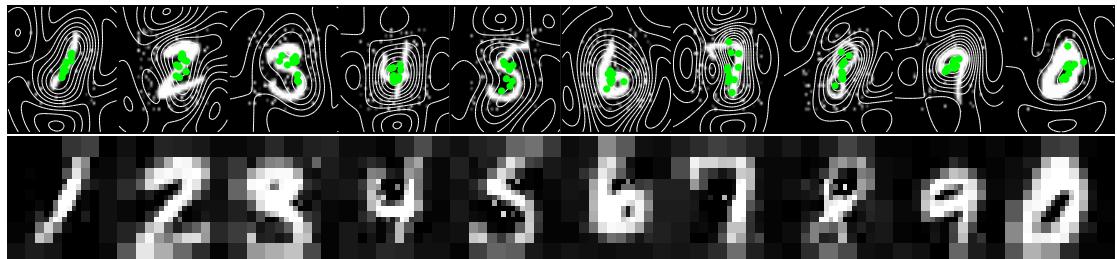


Figure 10: **Top:** Digit templates with the estimated reward surfaces superimposed. Markers indicate the best fixation point found in each of ten runs. **Bottom:** A visualization of the image found by averaging the best fixation points found across ten runs.

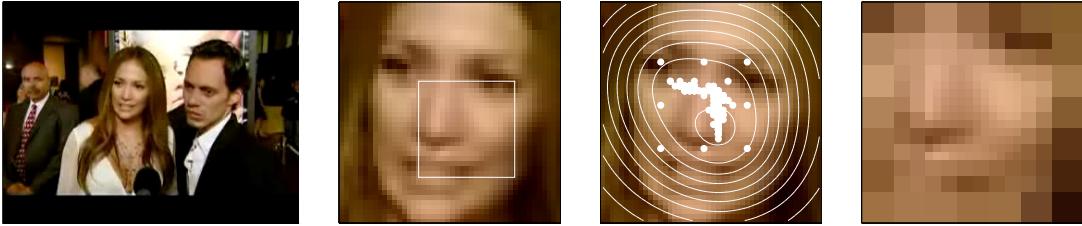


Figure 11: *Results on a real data set.* **Far left:** An example frame from the video sequence. **Center left:** The tracking template with the optimal fixation window highlighted. **Center right:** The reward surface produced by Bayesian optimization. The white markers show the centers of each fixation point in a single tracking run. **Right:** Input to the observation model when fixating on the best point. (Best viewed from a distance).

rithms could be used to train the RBMs online. Following the same methodology, we should also be able to adapt and improve the target templates and proposal distributions over time. This is essential to extend the results to long video sequences where the object undergoes significant transformations (e.g. as is done in the predator tracking system (Kalal et al., 2010)).

Deployment to more complex video sequences will require more careful and thoughtful design of the proposal distributions, transition distributions, control algorithms, template models, data-association and motion analysis modules. Fortunately, many of the solutions to these problems have already been engineered in the computer vision, tracking and online learning communities. Admittedly, much work remains to be done.

Saliency maps are ubiquitous in visual attention studies. Here, we simply used standard saliency tools and motion flow in the construction of the proposal distributions for particle filtering. There might be better ways to exploit the saliency maps, as neurophysiological experiments seem to suggest (Gottlieb et al., 1998).

One of the most interesting avenues for future work is the construction of more abstract attentional strategies. In this work, we focused on attending to regions of the visual field, but clearly one could attend to subsets of receptive fields or objects in the deep appearance model.

The current model has no ability to recover from a tracking failure. It may be possible to use information from the identity pathway (e.g. the classifier output) to detect and recover from tracking failure.

A closer examination of the exploration/exploitation tradeoff in the tracking setting

is in order. For instance, the methods we considered assume that future rewards are independent of past actions. This assumption is clearly not true in our setting, since choosing a long sequence of very poor fixation points can lead to tracking failure. We can potentially solve this problem by incorporating the current tracking confidence into the gaze selection strategy. This would allow the exploration/exploitation trade off to be explicitly modulated by the needs of the tracker, e.g. after choosing a poor fixation point the selection policy could be adjusted temporarily to place extra emphasis on exploiting good fixation points until confidence in the target location has been recovered.

Acknowledgments

We thank Ben Marlin, Kenji Okuma, Marc'Aurelio Ranzato and Kevin Swersky. This work was supported by CIFAR's NCAP program and NSERC.

References

- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R.E. Gambling in a rigged casino: The adversarial multi-armed bandit problem. Technical report, 1998a.
- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2001.
- Auer, Peter, Cesa-Bianchi, Nicolò, Freund, Yoav, and Schapire, Robert E. Gambling in a rigged casino: the adversarial multi-armed bandit problem. Technical Report NC2-TR-1998-025, 1998b.
- Barrington, L., Marks, T. K., Hsiao, J. H., and Cottrell, G. W. Nimble: a kernel density model of saccade-based visual memory. *Journal of Vision*, 8(14):1–14, 2008.
- Bazzani, L., de Freitas, N., Larochelle, H., Murino, V., and Ting, J.A. Learning attentional policies for tracking and recognition in video with deep networks. In Getoor, Lise and Scheffer, Tobias (eds.), *International Conference on Machine Learning*, pp. 937–944. ACM, 2011.

Brochu, E., Cora, V.M., and de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-23, University of British Columbia, 2009.

Butko, Nicholas J. and Movellan, Javier R. I-POMDP: An infomax model of eye movement. In *Proceedings of the International Conference on Development and Learning (ICDL 2008)*, August 2008.

Colombo, John. The development of visual attention in infancy. *Annual Review of Psychology*, pp. 337–367, 2001.

Doucet, A, de Freitas, N, and Gordon, N. Introduction to sequential Monte Carlo methods. In Doucet, A, de Freitas, N, and Gordon, N J (eds.), *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

Erez, Tom, Tramper, Julian J., Smart, William D., and Gielen, Stan C. A. M. A pomdp model of eye-hand coordination. In Burgard, Wolfram and Roth, Dan (eds.), *25th Conference on Artificial Intelligence (AAAI 2011)*, pp. 952–957. AAAI Press, 2011.

Freund, Y and Haussler, D. Unsupervised learning of distributions of binary vectors using 2-layer networks. In *Advances in Neural Information Processing Systems 4 (NIPS 4)*, pp. 912–919. Morgan Kaufman Publishers, 1991.

Freund, Yoav and Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

Gaborski, R., Vaingankar, V., Chaoji, V., Teredesai, A., and Tentler, A. Detection of inconsistent regions in video streams. In *Proc. SPIE Human Vision and Electronic Imaging*. Citeseer, 2004.

Gao, D., Mahadevan, V., and Vasconcelos, N. The discriminant center-surround hypothesis for bottom-up saliency. *Advances in neural information processing systems*, 20, 2007.

- Girard, B. and Berthoz, A. From brainstem to cortex: Computational models of saccade generation circuitry. *Progress in Neurobiology*, 77(4):215 – 251, 2005.
- Goodale, M.A. and Milner, A.D. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992.
- Gottlieb, Jacqueline P., Kusunoki, Makoto, and Goldberg, Michael E. The representation of visual salience in monkey parietal cortex. *Nature*, 391:481–484, 1998.
- Hinton, GE and Salakhutdinov, RR. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.
- Hinton, Geoffrey. E. A practical guide to training restricted Boltzmann machines. Technical Report UTML TR 2010-003, Department of Computer Science, University of Toronto, 2010.
- Hoffman, M., Brochu, E., and de Freitas, N. Portfolio allocation for Bayesian optimization. In *Uncertainty in Artificial Intelligence*, pp. 327–336. AUAI Press, 2011.
- Isard, M and Blake, A. Contour tracking by stochastic propagation of conditional density. In *European Computer Vision Conference*, pp. 343–356, 1996.
- Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (11):1254 –1259, 1998.
- Jones, D.R., Perttunen, C.D., and Stuckman, B.E. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.
- Kaelbling, L.P., Littman, M.L., and Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.
- Kalal, Z., Mikolajczyk, K., and Matas, J. Face-tld: Tracking-learning-detection applied to faces. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 3789–3792. IEEE, 2010.

Kanan, Christopher and Cottrell, Garrison W. Robust classification of objects, faces, and flowers using natural image statistics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pp. 2472–2479, 2010.

Kavukcuoglu, K., Ranzato, M.A., Fergus, R., and Le-Cun, Yann. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition*, pp. 1605–1612, 2009.

Kim, M., Kumar, S., Pavlovic, V., and Rowley, H. Face tracking and recognition with visual constraints in real-world videos. pp. 1–8, 2008.

Koch, C. and Ullman, S. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol*, 4(4):219–27, 1985.

Köster, Urs and Hyvärinen, Aapo. A two-layer ICA-like model estimated by score matching. In *International Conference of Artificial Neural Networks*, pp. 798–807, 2007.

Lampert, C. H., Blaschko, M. B., and Hofmann, T. Beyond sliding windows: Object localization by efficient subwindow search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

Larochelle, Hugo and Hinton, Geoffrey. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Neural Information Processing Systems*, 2010.

Lee, H., Grosse, R., Ranganath, R., and Ng, A.Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, 2009.

Martinez-Cantin, R., de Freitas, N., Doucet, A., and Castellanos, J. Active policy learning for robot planning and exploration under uncertainty. In *Proceedings of Robotics: Science and Systems*, 2007.

McNaughton, Bruce L., Battaglia, Francesco P., Jensen, Ole, Moser, Edvard I., and Moser, May-Britt. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678, 2006.

Najemnik, Jiri and Geisler, Wilson S. Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391, March 2005.

Nelson, J.D., McKenzie, C.R.M., Cottrell, G.W., and Sejnowski, T.J. Experience matters: Information acquisition optimizes probability gain. *Psychological science*, 21(7):960–969, 2010.

Okuma, Kenji, Taleghani, Ali, de Freitas, Nando, and Lowe, David G. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

Olshausen, B. A. and Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

Olshausen, B.A., Anderson, C.H., and Van Essen, D.C. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *The Journal of Neuroscience*, 13(11):4700, 1993a. ISSN 0270-6474.

Olshausen, Bruno A., Anderson, Charles H., and Essen, David C. Van. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13:4700–4719, 1993b.

O'Reilly, R. The what and how of prefrontal cortical organization. *Trends in neurosciences*, 33(8):355–361, 2010.

Postma, Eric O., van den Herik, H. Jaap, and Hudson, Patrick T. W. SCAN: A scalable model of attentional selection. *Neural Networks*, 10(6):993 – 1015, 1997.

Ranzato, M.A. and Hinton, G.E. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Computer Vision and Pattern Recognition*, pp. 2551–2558, 2010.

Rasmussen, C.E. and Williams, C.K.I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

Rensink, Ronald A. The dynamic representation of scenes. *Visual Cognition*, pp. 17–42, 2000.

Rosa, M.G.P. Visual maps in the adult primate cerebral cortex: Some implications for brain development and evolution. *Brazilian Journal of Medical and Biological Research*, 35:1485 – 1498, 2002.

Smolensky, Paul. Information Processing in Dynamical Systems: Foundations of Harmony Theory. volume 1, chapter 6, pp. 194–281. MIT Press, Cambridge, 1986.

Srinivas, N., Krause, A., Kakade, S.M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. pp. 1015–1022, 2010.

Swersky, K., Buchman, D., Marlin, B.M., and de Freitas, N. On autoencoders and score matching for energy based models. *International Conference in Machine Learning*, 2011.

Taylor, G.W., Sigal, L., Fleet, D.J., and Hinton, G.E. Dynamical binary latent variable models for 3D human pose tracking. In *Computer Vision and Pattern Recognition*, pp. 631–638, 2010.

Torralba, A., Oliva, A., Castelhano, M.S., and Henderson, J.M. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological review*, 113(4):766, 2006.

Ungerleider, L. and Mishkin, M. Two cortical visual systems. pp. 549–586. MIT Press, 1982.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pp. 1096–1103, 2008.

Vogel, J. and de Freitas, N. Target-directed attention: Sequential decision-making for gaze planning. In *IEEE International Conference on Robotics and Automation*, pp. 2372 –2379, 2008.

Welling, M., Rosen-Zvi, M., and Hinton, G. Exponential family harmoniums with an application to information retrieval. *Neural Information Processing Systems*, 17: 1481–1488, 2005.

Zhang, L., Tong, M.H., and Cottrell, G.W. Sunday: Saliency using natural statistics for dynamic analysis of scenes. In *Proceedings of the 31st Annual Cognitive Science Conference, Amsterdam, Netherlands*. Citeseer, 2009.